

BAB 5

SIMPULAN DAN SARAN

Bab ini berisikan tentang simpulan dan saran terkait dengan pekerjaan yang telah dilakukan dan dijelaskan pada Bab terdahulu.

5.1 Simpulan

Berdasarkan penelitian yang sudah dilakukan, didapat kesimpulan sebagai berikut:

1. Model *Self-Attention for Knowledge Tracing* (SAKT) berhasil diimplementasikan kepada sebuah sistem *Intelligent Tutoring System* (ITS) untuk melakukan fungsi *adaptive learning*. Implementasi berhasil karena model SAKT dapat memberikan prediksi probabilitas yang sesuai dengan riwayat pelajar agar aplikasi ITS dapat mengadaptasikan soal yang diberikan kepada pelajar sesuai dengan performa dari pelajar tersebut. Pelajar yang performa yang baik akan diberikan soal yang lebih sulit dalam waktu yang lebih singkat, sedangkan pelajar dengan performa yang kurang baik diberikan soal-soal dengan tingkat kesulitan yang meningkat secara perlahan. Selain itu, aplikasi juga berfungsi sesuai ekspektasi berdasarkan 16 skenario pengujian pada *blackbox testing* dengan tingkat keberhasilan 16 dari 16 skenario atau 100%.
2. Setelah melakukan *hyperparameter tuning* untuk model SAKT, sebanyak 16 model berhasil dibangun dan dievaluasi. Namun berdasarkan metrik evaluasi yang digunakan dalam model SAKT, yaitu skor *Area Under the Receiving Operating Characteristic Curve* (AUC), ditemukan bahwa *hyperparameter* tidak memiliki pengaruh yang signifikan terhadap model SAKT, terutama yang dilatih menggunakan dataset ASSISTments 2015 Skill Builder dengan jumlah soal yang ada pada *dataset* dan jumlah soal yang dapat diakomodasi oleh model SAKT adalah 100. Dari 16 model tersebut, skor AUC yang didapat berkisar sekitar angka 0.7122 dengan skor AUC tertinggi adalah 0.7154. Skor AUC tertinggi pada 0.7154 didapat dengan menggunakan *hyperparameter* $max_length = 200$, $batch_size = 200$, $embed_size = 100$, $num_attn_layers = 2$, $drop_prob = 0.1$, dan $lr = 0.002$. Skor AUC yang didapat dari pembangunan model awal dan hasil *hyperparameter tuning* tersebut

menunjukkan bahwa model dapat merepresentasikan probabilitas di dunia nyata dengan cukup baik.

5.2 Saran

Berdasarkan penelitian yang sudah dilakukan, saran untuk penelitian/pengembangan selanjutnya adalah sebagai berikut:

1. Model SAKT dapat dilatih dengan *dataset* lain dengan fitur yang lebih lengkap dan jumlah baris data yang lebih banyak untuk meningkatkan performa model dan melakukan prediksi probabilitas dengan lebih baik. Contoh *dataset* lain yang dapat digunakan adalah data ASSISTments pada tahun lainnya (2009-2010, 2012-2013, atau 2017) yang memiliki tambahan fitur seperti *timestamp*/tanggal serta jam pengerjaan dan jumlah waktu yang dibutuhkan untuk mengirimkan jawaban. Selain itu, jumlah interaksi/total baris data yang lebih banyak juga dapat meningkatkan performa serta kemampuan generalisasi model.
2. Dengan menggunakan *dataset* ASSISTments 2015 Skill Builder, aplikasi memiliki limitasi pada jumlah soal pada setiap materi. Hal ini dikarenakan model hanya dapat melakukan perhitungan sesuai dengan jumlah soal yang ada pada *dataset*. Apabila model menemukan soal yang diluar atau melebihi jumlah soal yang ada pada *dataset*, maka model akan menolak pemrosesan dan prediksi tidak dapat dilakukan. Oleh karena itu untuk pengembangan selanjutnya, dapat digunakan *dataset* dengan jumlah soal yang lebih banyak agar model SAKT dapat mengakomodasi jumlah soal yang lebih banyak juga dalam melakukan prediksi probabilitas.
3. Sistem ini dapat diintegrasikan dengan aplikasi *tutoring* yang sudah operasional, terutama aplikasi yang menyimpan semua riwayat pembelajaran pelajarannya secara lengkap. Data riwayat tersebut dapat digunakan sebagai *dataset* untuk melatih model SAKT yang baru dan mengintegrasikan model SAKT sebagai implementasi fungsi adaptasi soal yang dibangun pada penelitian ini ke sistem *tutoring* yang sudah operasional.
4. Aplikasi yang dibangun pada penelitian ini hanya terfokus pada sisi pelajar. Oleh karena itu, pengembangan selanjutnya juga dapat membuat aplikasi tambahan bagi sisi *admin* atau penyedia layanan *tutoring* untuk

mengelola aplikasi. Pengembangan sisi *admin* dapat dilakukan dengan membangun aplikasi terpisah yang berbasis web (misalnya menggunakan React + Node.js) untuk memungkinkan manajemen soal, materi, dan data pelajar secara fleksibel tanpa mengakses *database* langsung. Aplikasi sisi *admin*/penyelenggara akan memudahkan pengelolaan dan pemeliharaan sistem secara berkelanjutan

